Byzantine-Resilient Federated Principal Subspace Estimation

2024 IEEE International Symposium on Information Theory

Ankit Pratap Singh and Namrata Vaswani

Iowa State University

Problem Setting

Estimate principal subspace $span(\mathbf{U}^*)$ of an unknown $n \times n$ symmetric matrix $\mathbf{\Phi}^*$ in a federated setting, while being resilient to **Byzantine Attacks**.

$$\mathbf{D}_{n\times q} = [(\mathbf{D}_1)_{n\times q_1}, \ldots, (\mathbf{D}_\ell)_{n\times q_\ell}, \ldots, (\mathbf{D}_L)_{n\times q_L}]$$

1. $\mathbf{U}^*_{n\times r}$ denotes the top r eigenvectors of $\mathbf{\Phi}^*$.
2. **Federated Setting:** Each node $\ell \in [L]$ observes a data matrix $\mathbf{D}_\ell$, that allows it
   - To estimate $\mathbf{\Phi}^*$ as $\mathbf{\Phi}_\ell = \mathbf{D}_\ell \mathbf{D}_\ell^\top / q_\ell$
   - To estimate $\mathbf{U}^*$ as $\mathbf{U}_\ell$, which are the top r eigenvectors of $\mathbf{\Phi}_\ell$

Applications

- Byzantine Resilient PCA, subspace tracking.
- To make the spectral initialization step of many Low Rank matrix recovery problems secure in a federated setting.
- In particular we have done it for LRCS a.k.a Multi Task Representation Learning or Few Shot Learning. See our **ICML 24 paper "Byzantine Resilient and Fast Federated Few-Shot Learning"**

Byzantine Attacks [Mhamdi, Guerraoui, and Rouault, ICML, 2018][1]

Byzantine attack is a "model update poisoning" attack where

1. It knows the full state of the center and every node (data and algorithm, including all algorithm parameters).
2. Different Byzantine adversaries can also collude.
3. They cannot modify the outputs of the other (non-Byzantine) nodes or of the center, or delay communication.

Byzantine nodes can thus design the worst possible attacks at each algorithm iteration.

[1] Mhamdi, Guerraoui, and Rouault, The hidden vulnerability of distributed learning in byzantium

Dealing with Attacks: Geometric Median (GM)

- For L data vectors $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L$, this is defined as $\mathbf{z}_{gm} = \min_{\mathbf{z}} \sum_{\ell=1}^{L} \|\mathbf{z}_\ell - \mathbf{z}\|_2$.
- The GM cannot be computed in closed form.
- Two popular iterative solutions for computing this are Weiszfeld's algorithm [2] and Accurate Median [3]. The most commonly used in practice is Weiszfeld's algorithm.

[2] Endre Weiszfeld, Sur le point pour lequel la somme des distances den points donnés est minimum

[3] Cohen et al., Geometric median in nearly linear time

GM Theorem: Let $\{\mathbf{z}_1, ..., \mathbf{z}_L\}$ with each $\mathbf{z}_\ell \subseteq \Re^n$ denote L nodes output, and let $\mathbf{z}_{gm}$ denote exact Geometric Median. Assume 60% satisfy

$$\|\mathbf{z}_\ell - \tilde{\mathbf{z}}\| \leq \epsilon$$

then,

$$\|\mathbf{z}_{gm} - \tilde{\mathbf{z}}\| \leq 6\epsilon$$

Including probability argument

$$\Pr\{\|\mathbf{z}_\ell - \tilde{\mathbf{z}}\| \leq \epsilon\} \geq 1 - p$$

Then, w.p. at least $1 - \exp(-L\psi(0.4 - \tau, p))$,

$$\|\mathbf{z}_{gm} - \tilde{\mathbf{z}}\| \leq 6\epsilon$$

Subspace Distance [Chen et al., Foundations and Trends® in Machine Learning, 2021][4]

- One can represent $r-$dimensional subspace by a matrix $\mathbf{U}^*$, whose columns form an orthonormal basis of the subspace.
- Projection matrix for subspace $\mathbf{U}^*$

$$\mathcal{P}_{\mathbf{U}^*} := \mathbf{U}^*\mathbf{U}^{*\top}$$

- Subspace Distance (SD) between $span(\mathbf{U})$, $span(\mathbf{U}^*)$

$$\mathbf{SD}_F(\mathbf{U}, \mathbf{U}^*) := \|(\mathbf{I} - \mathbf{U}\mathbf{U}^\top)\mathbf{U}^*\|_F$$

[4] Chen et al., Spectral methods for data science: A statistical perspective

Subspace-Median: Key Ideas

- The GM is defined for quantities whose distance can be measured using the vector l_2 norm (equivalently, matrix Frobenius norm).
- Our solution adapts the GM to use it for subspaces by using the fact that the Frobenius norm between the projection matrices of two subspaces is another measure of subspace distance: $\|\mathcal{P}_{\mathbf{U}} - \mathcal{P}_{\mathbf{U}^*}\|_F = \sqrt{2}\mathbf{SD}_F(\mathbf{U}, \mathbf{U}^*)$.

Algorithm [Singh & Vaswani, IEEE ISIT, 2024][5]

Algorithm 1 Subspace Median

1: **Nodes** $\ell = 1, ..., L$

2: Estimates $\mathbf{U}_\ell$, which are the top r eigenvectors of $\mathbf{\Phi}_\ell = \mathbf{D}_\ell \mathbf{D}_\ell^\top / q_\ell$

3: Sends $\mathbf{U}_\ell$ which are of size $n \times r$ to Central Server

4: **Central Server**

5: Orthonormalize: $\mathbf{U}_\ell \leftarrow QR(\hat{\mathbf{U}}_\ell)$, $\ell \in [L]$

6: Compute $\mathcal{P}_{\mathbf{U}_\ell} \leftarrow \mathbf{U}_\ell \mathbf{U}_\ell^\top$, $\ell \in [L]$

7: Compute GM: $\mathcal{P}_{gm} \leftarrow GM\{\mathcal{P}_{\mathbf{U}_\ell}, \ell \in [L]\}$

8: Find $\ell_{best} = \arg\min_\ell \|\mathcal{P}_{\mathbf{U}_\ell} - \mathcal{P}_{gm}\|_F$

9: Output $\mathbf{U}_{out} = \mathbf{U}_{\ell_{best}}$

[5] Singh & Vaswani, Byzantine-Resilient Federated Principal Subspace Estimation

Subspace-Median

Lemma

Suppose GM can be computed exactly and at least 60% $\mathbf{U}_\ell$*'s satisfy*

$$\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_\ell) \leq \delta$$

then,

$$\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_{out}) \leq 23\delta$$

- Including **probability argument**, If

$$\Pr\left(\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_\ell) \leq \delta\right) \geq 1 - p$$

then,

$$\Pr\left(\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_{out}) \leq 23\delta\right) \geq 1 - \exp(-L\psi(0.4 - \tau, p))$$

$$\psi(a, b) := (1 - a)\log\frac{1 - a}{1 - b} + a\log\frac{a}{b}$$

- If GM is approximated using using a linear time algorithm[6] then,

$$\Pr\left(\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_{out}) \leq 23\delta\right) \geq 1 - \mathbf{c_0} - \exp(-L\psi(0.4 - \tau, p))$$

[6] Cohen et al., Geometric median in nearly linear time

Theorem (Subspace-Median)

Consider Subspace-Median Algorithm. For a $\tau < 0.4$, *suppose that, for at least* $(1-\tau)L$ $\mathbf{U}_\ell$*'s*

$$\Pr\left(\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_\ell) \leq \delta\right) \geq 1 - p$$

then, with probability at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, p))$,

$$\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_{out}) \leq 23\delta.$$

Proof Sketch

- **GM Theorem:**

$$\Pr\{\|\mathbf{z}_\ell - \tilde{\mathbf{z}}\| \leq \epsilon\} \geq 1 - p$$

Then,

$$\Pr\{\|\mathbf{z}_{gm} - \tilde{\mathbf{z}}\| \leq 6\epsilon\} \geq 1 - \exp(-L\psi(0.4 - \tau, p))$$

- Since $\mathbf{SD}_F(\mathbf{U}_\ell, \mathbf{U}^*) = (1/\sqrt{2})\,\|\mathcal{P}_{\mathbf{U}_\ell} - \mathcal{P}_{\mathbf{U}^*}\|_F$, thus,

$$\|\mathcal{P}_{\mathbf{U}_\ell} - \mathcal{P}_{\mathbf{U}^*}\|_F \leq \sqrt{2}\delta$$

for at least 60% of $\mathbf{U}'_\ell$s.

- Using GM theorem, we have w.p. at least $1 - \exp\left(-L\psi\left(0.4 - \tau, p\right)\right)$

$$\|\mathcal{P}_{gm} - \mathcal{P}_{\mathbf{U}^*}\|_F \leq 6\sqrt{2}\delta$$

- We use this to bound the **SD** between $\mathbf{U}_{\ell_{best}}$ and $\mathbf{U}^*$ by using simple algebra.

Resilient Federated PCA

$$\mathbf{D} := [\mathbf{d}_1, \mathbf{d}_2, \dots \mathbf{d}_q] = [(\mathbf{D}_1)_{n\times q_1}, ..., (\mathbf{D}_\ell)_{n\times q_\ell}, ..., (\mathbf{D}_L)_{n\times q_L}]$$

Given q data vectors $\mathbf{d}_k \in \Re^n$, that are

- *Zero Mean*
- Mutually Independent
- $K-$Sub-Gaussian
- Have covariance matrices that share the same principal subspace

$$\mathbf{\Sigma}_k^* \overset{\text{EVD}}{=} [\mathbf{U}^*, \mathbf{U}_{\perp,k}^*]\mathcal{S}_k[\mathbf{U}^*, \mathbf{U}_{\perp,k}^*]^\top$$

The goal is to obtain a resilient estimate of the r-dimensional subspace $\mathbf{U}^*$ of $\Re^n$ in a federated setting. L nodes out of which $L_{byz} = \tau L$ are Byzantine.

Resilient Federated PCA via Subspace Median

Corollary (Subspace Median for PCA)

Consider the PCA problem. Assume at most τL of the L total nodes are Byzantine, with $\tau \leq 0.4$ and that $\sigma_r^* - \sigma_{r+1}^* \geq \Delta$ for a $\Delta > 0$. If

$$q_\ell := \frac{q}{L} \geq CK^4 \frac{\sigma_1^{*2}}{\Delta^2} \cdot \frac{nr}{\epsilon^2},$$

then, w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, 2\exp(-n) + n^{-10}))$,

$$\mathbf{SD}_F(\mathbf{U}_{out}, \mathbf{U}^*) \leq \epsilon$$

- The communication cost is nr per node.
- The computational cost at the center is order $n^2 L \log^3\left(\frac{Lr}{\epsilon}\right)$.
- The computational cost at the node is order $nq_\ell r \frac{\sigma_r^*}{\Delta} \log(\frac{n}{\epsilon})$.

Resilient Federated PCA via Subspace Median of Means

In order to implement the "mean" step, we combine samples from $\rho = \frac{L}{\tilde{L}}$ ($\tilde{L} < L$) nodes by implementing $\tilde{L}$ different federated power methods.

Corollary

Assume that the set of Byzantine nodes remains fixed for all iterations and the size of this set is at most τL with $\tau < 0.4\tilde{L}/L$. If

$$\frac{q}{L} = \widetilde{q} \geq CK^4 \frac{\sigma_1^{*2}}{\Delta^2} \frac{nr}{\epsilon^2} \cdot \frac{\tilde{L}}{L}$$

then, then, w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, 2\exp(-n) + n^{-10}))$,

$$\mathbf{SD}_F(\mathbf{U}_{out}, \mathbf{U}^*) \leq \epsilon$$

Comparisons for solving the resilient federated PCA problem

Methods$\rightarrow$	**SubsMed**	**ResPowMeth**	**SVD-RCE**
Samp. Comp. (bound on q)	$\frac{nrL}{\epsilon^2}$	$\max\left(n^2r^2, \frac{n}{\epsilon^2}\right) \cdot L$	$\frac{n^2L}{\epsilon^2}$
Comm Cost	nr	$nr\frac{\sigma_r^*}{\Delta}\log(\frac{n}{\epsilon})$	n^2
Cost - node	$nq_\ell r\frac{\sigma_r^*}{\Delta}\log(\frac{n}{\epsilon})$	$nq_\ell r\frac{\sigma_r^*}{\Delta}\log(\frac{n}{\epsilon})$	n^2q_ℓ
Cost - center	$n^2L\log^3\left(\frac{Ln}{\epsilon}\right)$	$nrL\frac{\sigma_r^*}{\Delta}\log(\frac{n}{\epsilon})\log^3\left(\frac{Ln}{\epsilon}\right)$	$n^2L\log^3\left(\frac{Ln}{\epsilon}\right)$

- Resilient Power Method (ResPowMeth): GM based modification of the power method[7]
- SVD-Resilient Covariance Estimation (SVD-RCE): SVD on GM of Covariance matrices[8]

[7] Hardt and Price, The noisy power method: A meta algorithm with applications
[8] Minsker, Geometric median and robust estimation in Banach spaces

Resilient Federated PCA Experiment

Attacks	SubsMed (Proposed)	ResPowMeth	PowMeth (No Attack)
Alternating	0.091	0.898	0.050
Ones	0.091	0.952	0.050
Orthogonal	0.091	0.208	0.050

Table 1: $n = 1000,\ L = 3,\ L_{byz} = 1,\ r = 60,\ \widetilde{q} = 600,$ rank-$(r+1)$

Resilient Federated PCA Experiment

Attacks	SubsMed (Proposed)	ResPowMeth	PowMeth (No Attack)
Alternating	0.348	0.972	0.182
Ones	0.349	0.990	0.182
Orthogonal	0.348	0.366	0.182

Table 2: $n = 1000$, $L = 3$, $L_{byz} = 1$, $r = 60$, $\widetilde{q} = 600$, full rank

Low Rank Columnwise Compressive Sensing (Multi Task Representation Learning) (Few-Shot Learning)

$$\mathbf{y}_k = \mathbf{X}_k \theta_k^*, k \in [q]$$

$$\mathbf{\Theta}^* = \mathbf{U}^* \mathbf{B}^*$$

Solving this problem requires solving

$$\min_{\substack{\tilde{\mathbf{U}} \in \Re^{n \times r} \\ \tilde{\mathbf{B}} \in \Re^{r \times q}}} \sum_{k=1}^{q} \left\| \mathbf{y}_k - \mathbf{X}_k \tilde{\mathbf{U}} \tilde{\mathbf{b}}_k \right\|^2 \tag{1}$$

AltGDmin[9], a fast and communication-efficient GD-based algorithm was introduced for solving the mathematical problem given in (1)

[9] Nayer & Vaswani, Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections

AltGDmin [10]

- Initialization step Each node computes $(\mathbf{U}_0)_\ell$ which is the matrix of top r left singular vectors of

$$(\hat{\mathbf{\Theta}}_0)_\ell := \sum_{k=1}^{q} (\mathbf{X}_k)_\ell{}^\top ((\mathbf{y}_k)_\ell)_{\text{trunc}} \mathbf{e}_k^\top$$

  - $\mathbf{U}_0 =$ SubspaceMedian of $(\mathbf{U}_0)_\ell$

- Alt-GD-Min: at each iteration $t \geq 1$, alternate b/w
  - min for $\mathbf{B}$

$$\mathbf{B} \leftarrow \arg\min_{\tilde{\mathbf{B}}} f(\mathbf{U}, \tilde{\mathbf{B}}) \;\Leftrightarrow \mathbf{b}_k = (\mathbf{X}_k \mathbf{U})^\dagger \mathbf{y}_k, \;\; k \in [q]$$

  - projected GD for $\mathbf{U}$

$$\mathbf{U}^+ \leftarrow \text{QR}(\mathbf{U} - \eta \text{GeometricMedian}\{\nabla_U f_\ell(\mathbf{U}, \mathbf{B})\}_{\ell=1}^L)$$

$$\mathbf{U} \leftarrow \mathbf{U}^+$$

[10] Nayer & Vaswani, Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections

Algorithm 2 Byz-Fed-AltGDmin-Learn: Complete algorithm

Nodes $\ell = 1, ..., L$

Compute $(\mathbf{U}_0)_\ell$ which is the matrix of top r left singular vectors of $(\hat{\mathbf{\Theta}}_0)_\ell := \sum_{k=1}^{q} (\mathbf{X}_k)_\ell{}^\top ((\mathbf{y}_k)_\ell)_{\mathrm{trunc}} \mathbf{e}_k^\top$

Central Server (implements Subspace Median on $(\mathbf{U}_0)_\ell$, $\ell \in [L]$)

Orthonormalize: $\mathbf{U}_\ell \leftarrow QR((U_\ell)_0)$, $\ell \in [L]$

Compute $\mathcal{P}_{\mathbf{U}_\ell} \leftarrow \mathbf{U}_\ell \mathbf{U}_\ell^\top$, $\ell \in [L]$

Compute GM: $\mathcal{P}_{gm} \leftarrow \mathrm{GM}\{\mathcal{P}_{\mathbf{U}_\ell}, \ell \in [L]\}$

Find $\ell_{best} = \arg\min_\ell \|\mathcal{P}_{\mathbf{U}_\ell} - \mathcal{P}_{gm}\|_F$

Output $\mathbf{U}_0 = \mathbf{U}_{out} = \mathbf{U}_{\ell_{best}}$

for $t = 1$ to T **do**

  Nodes $\ell = 1, ..., L$

  Set $\mathbf{U} \leftarrow \mathbf{U}_{t-1}$

  $(\mathbf{b}_k)_\ell \leftarrow ((\mathbf{X}_k)_\ell \mathbf{U})^\dagger (\mathbf{y}_k)_\ell, \ \forall\ k \in [q]$

  $\nabla f_\ell \leftarrow \sum_{k \in [q]} (\mathbf{X}_k)_\ell^\top ((\mathbf{X}_k)_\ell \mathbf{U} (\mathbf{b}_k)_\ell - (\mathbf{y}_k)_\ell)(\mathbf{b}_k)_\ell{}^\top$

  Central Server

  $\nabla f^{GM} \leftarrow \mathrm{GM}(\nabla f_\ell, \ell = 1, 2, \ldots L)$.

  Compute $\mathbf{U}^+ \leftarrow QR(\mathbf{U}_{t-1} - \frac{\eta}{\rho \widetilde{m}} \nabla f^{GM})$

  return Set $\mathbf{U}_t \leftarrow \mathbf{U}^+$. Push $\mathbf{U}_t$ to nodes.

end for

Multi-task representation learning/Few-shot Learning

"Byzantine Resilient and Fast Federated Few-Shot Learning" using subspace median will be presented at ICML 2024.

Theorem

(Byz-Fed-AltGDmin-Learn: Complete guarantee) Assume $\max_k \|\mathbf{b}_k^*\| \leq \mu\sqrt{r/q}\sigma_1(\mathbf{\Theta}^*)$ *for a constant* $\mu \geq 1$. *If*

$$\frac{m}{L}q \geq C\kappa^4\mu^2(n+q)r^2\log(1/\epsilon)$$

then, w.p. at least $1 - TLn^{-10}$,

$$\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_T) \leq \epsilon$$

and $\|(\theta_k)_\ell - \theta_k^*\| \leq \epsilon\|\theta_k^*\|$ *for all* $k \in [q]$, $\ell \in [L]$. *The communication cost per node is order* $nr\log(\frac{n}{\epsilon})$. *The computational cost at any node is order* $nqr\log(\frac{n}{\epsilon})$ *while that at the center it is* $n^2L\log^3(Lr/\epsilon)$.

Thank You!